

In re application of CABRERA ET AL.
Serial No. 09/360,542

REMARKS

The Office action has been carefully considered. Claims 56-95 were pending in the application. The Office action rejected claims 56-70, 75-84, 86-92, and 95 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,924,102 to Perks et al ("Perks"). The Office action rejected claims 71-74, 85, and 93-94 under 35 U.S.C. § 103(a) over Perks in view of U.S. Patent 5,469,573 to McGill, III et al ("McGill"). Applicants respectfully traverse these rejections.

Applicants thanks the Examiner for the interview held (by telephone) on June 2, 2004. During the interview, the Examiner and applicant's attorney discussed the claims with respect to the prior art. The essence of applicant's position is incorporated in the remarks below.

Applicants' technique is generally directed towards automatic backup and restoration of a failed system by capturing and storing the state of executing programs in addition to backing up the actual operating system and data files. Applicants' technique provides an application programming interface that a backup program may invoke for capturing the state of executing programs, including application programs. FIG. 2 illustrates a backup process 60 for invoking backup APIs 80 for collecting information on running programs/processes 82 and storing the state information in state files 62. Applicants' technique may provide a backup application programming interface for collecting information on programs and processes in use on the system and whose state is changing. See page 17 of the specification. The information of a program's execution state may be persistently stored on a medium along with the

In re application of CABRERA ET AL.
Serial No. 09/360,542

state of a hardware device such as a hard disk, catalog information of unmodifiable files, and registry information. It should be noted that other information may be collected in a similar manner, thereby extending the backup and restore capabilities. For example, some programs may have their own database of information and the backup APIs may similarly back up this information on a running system. See page 19 of the specification. Upon system failure, a recovery process may retrieve the previous state of the hard disk persistently stored and restore the hard disk state. The recovery process may then retrieve the information of a previous execution state of programs from persistent storage and restore the programs. FIG. 3 illustrates a restore process 70 for retrieving state information on running programs/processes in state files 62 for use by a restore program 100 for restoration of applications 66. Running the restore application program 100 is the counterpart of the program used to back up the applications 66. See page 29 of the specification. Advantageously, the application programming interface may be used by any third party developer for backing up application programs for restoration upon failure. Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

Perks, in general, is very different from the present invention. First of all, Perks is directed to a system and method for managing critical files in an information handling system. Application programs may call an API to register and unregister critical files. Importantly, critical files are files that contain configuration data, setup information, or critical user data. Application

In re application of CABRERA ET AL.
Serial No. 09/360,542

executables are not considered critical files in Perks, nor does Perks disclose the concept of capturing and storing the state of executing programs as critical files. Perks distinguishes his invention from prior art methods that often back up only user data and do not back up configuration data. Perks' system manages and backs up critical files such as configuration files, setup files, and user data files, which are difficult to recover after a system crash. Nowhere in Perks does Perks suggest program execution information. Instead, Perks provides examples of initialization files and configuration files such as OS2SYS.INI, OS2.INI, CONFIG.SYS, AUTOEXEC.BAT, SYSTEM.DAT and USER.DAT. These are all conventional configuration, setup and user data files. As a further example, Perks also considers application-specific configuration and setup files as critical files. As an example, Perks includes .DOT files that store templates for Microsoft Word. As another example, Perks describes user data files such as for a financial application program. Nowhere does Perks disclose receiving information on the execution state of a program executing on the system. In fact, Perks teaches away from receiving information on the execution state of a program executing on the system by dismissing application executables as critical files.

In contrast to the claims of the present invention, Perks has simply no concept of storing the state of executing programs. Nor does Perks have any concept of providing application programming interface for capturing the state of executing programs as generally recited in applicants' claims. First, independent claim 56 recites the limitation of "an application programming interface for

In re application of CABRERA ET AL.
Serial No. 09/360,542

providing information of the software execution state." Applicants' technique may provide a backup application programming interface for collecting information on programs and processes in use on the system and whose state is changing. Perks does not disclose any such limitation, but is instead concerned with managing and backing up critical files such as configuration files, setup files, and user data files. Consistent with this stated purpose, Perks simply describes managing critical files in an information handling system whereby application programs may call an API to register and unregister critical files for backing up the critical files for later use during restoration of the computer system. Nowhere in Perks can there be found any discussion of providing an application programming interface for capturing software execution state.

Second, independent claim 65 of the present invention recites "receiving information of the execution state of at least one program executing on the system." Applicants' technique may collect information on programs and processes in use on the system and whose state is changing. Again, Perks does not disclose any such limitation. Rather, consistent with the stated purpose, Perks merely describes managing critical files in an information handling system whereby application programs may call an API to register and unregister critical files for backing up the critical files for later use during restoration of the computer system. Third, independent claim 81 recites "retrieving information of a previous execution state of the at least one program stored on a medium". Applicants' technique further uses the information of the execution state of a programs and processes to restore a computer system after failure. Perks does

In re application of CABRERA ET AL.
Serial No. 09/360,542

not disclose the limitation of claim 81. Moreover, Perks does not describe any capability to retrieve information of a previous execution state of a program, nor could Perks without previously collecting information on programs and processes in use on the system including their execution state. Nowhere in Perks can there be found any discussion of capturing software execution state or retrieving such a captured software execution state.

McGill is also very different from the present invention. First of all, McGill is directed to a data backup and restoration technique for restoring a configured operating system without reloading and reconfiguring the operating system from its original distribution media. To do so, McGill describes backing up the configured operating system by copying the operating system loaded on a computer system to a first media or backup device and copying configuration-specific data files such as system configuration files and device drivers to a second media or bootable recovery diskette. The configured operating system may then be subsequently restored on the computer system by initializing the computer system from the bootable recovery diskette and using the system configuration files and device drivers stored on the recovery diskette to provide a temporary operating system for loading a recovery application program from the backup device that, in turn, loads the operating system, configuration files and device drivers from the backup device. After loading is complete, the computer system is reinitialized using the loaded operating system. Once the computer system is restarted, any other backed up files can be restored to the hard drive.

In re application of CABRERA ET AL.
Serial No. 09/360,542

Like Perks, McGill has simply no concept of storing the state of executing programs. Nor does McGill have any concept of providing an application programming interface for capturing the state of executing programs as generally recited in applicants' claims. Rather McGill describes backing up the configured operating system by copying the operating system loaded on a computer system to a first media or backup device and copying configuration-specific data files such as system configuration files and device drivers to a second media or bootable recovery diskette. Nowhere in McGill can applicants find a description of storing the state of executing programs, nor can applicants find any concept of providing an application programming interface for capturing the state of executing programs as generally recited in applicants' claims.

For instance, independent claim 56 recites the limitation of "an application programming interface for providing information of the software execution state." Significantly, applicants' technique may provide a backup application programming interface for collecting information on programs and processes in use on the system and whose state is changing. McGill does not disclose any such limitation, but is instead concerned with restoring a configured operating system without reloading and reconfiguring the operating system from its original distribution media. Consistent with this stated purpose, McGill simply describes backing up the configured operating system by copying the operating system loaded on a computer system to a backup device and copying configuration-specific data files such as system configuration files and device drivers to a second media or bootable recovery diskette. Nowhere in McGill can there be

In re application of CABRERA ET AL.
Serial No. 09/360,542

found any discussion of providing an application programming interface for capturing software execution state.

As another example, independent claim 65 of the present invention recites "receiving information of the execution state of at least one program executing on the system." Applicants' technique may collect information on programs and processes in use on the system and whose state is changing. Again, McGill does not disclose any such limitation. Rather, consistent with the stated purpose, McGill merely describes backing up the configured operating system by copying the operating system loaded on a computer system to a backup device so that it may be restored without reloading and reconfiguring the operating system from its original distribution media. Moreover, independent claim 81 recites "retrieving information of a previous execution state of the at least one program stored on a medium". Applicants' technique further uses the information of the execution state of a programs and processes to restore the environment of a computer system after failure. McGill does not disclose the limitation of claim 81. Moreover, McGill does not describe any capability to retrieve information of a previous execution state of a program, nor could McGill without previously collecting information on programs and processes in use on the system including their execution state. Nowhere in McGill can there be found any discussion of capturing software execution state or retrieving such a captured software execution state.

**In re application of CABRERA ET AL.
Serial No. 09/360,542**

For at least these significant reasons, applicants submit that the claims are patentable over the prior art of record, including Perks and McGill, whether considered alone or in any permissible combination.

Further, by law, in order to modify a reference to reject claimed subject matter, there must be some teaching or suggestion outside of applicants' teachings to do so. Neither Perks nor McGill have any such teachings or suggestions as to any such modification, let alone any teaching or suggestion as to how either system could be modified, or why it might be desirable to do so. In specific, the motivation described by Perks for his technique is to manage and back up critical files such as configuration files, setup files, and user data files, which are difficult to recover after a system crash. Perks distinguishes his invention from prior art methods that often back up only user data and do not back up configuration data. Importantly, critical files are files that contain configuration data, setup information, or critical user data. Application executables are not considered critical files in Perks, nor does Perks disclose the concept of capturing and storing the state of executing programs as critical files. In fact, Perks teaches away from receiving information on the execution state of a program executing on the system by dismissing application executables as critical files. Nor does McGill offer any motivation for modifying his invention to reach applicants' claims. The motivation described by McGill for his technique is to restore a configured operating system without reloading and reconfiguring the operating system from its original distribution media. The only other use McGill contemplates for his technique is for efficient factory loading of a fully configured

In re application of CABRERA ET AL.
Serial No. 09/360,542

operating system and application software into similarly configured workstations on a production line (McGill, column 2, lines 6-17.) Thus, McGill offers no motivation for modifying his invention to reach applicants' recited limitations of "providing an application programming interface for capturing software execution state," or for "receiving information of the execution state of at least one program executing on the system." Thus, the only way in which Perks or McGill could be modified to reach applicants' claimed invention is via applicants' own teachings, which is impermissible by law.

For at least these additional reasons, applicants submit that the claims are patentable over the prior art of record, including Perks and McGill, whether considered alone or in any permissible combination.

Regarding the dependent claims, applicants respectfully submit that dependent claims 57-64, 66-80 and 82-95, by similar analysis, are not anticipated by Perks or McGill, whether considered alone or in any permissible combination. Each of the dependent claims 57-64 includes the limitation of "an application programming interface for providing information of the software execution state." Each of the dependent claims 66-80 includes the limitation of "receiving information of the execution state of at least one program executing on the system." And each of the dependent claims 82-95 includes the limitation of "retrieving information of a previous execution state of the at least one program stored on a medium." In addition to the limitations noted above, each of these dependent claims include additional patentable elements. For at least these

In re application of CABRERA ET AL
Serial No. 09/360,542

reasons, applicants submit that all of the dependent claims are in condition for allowance.

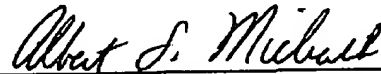
In re application of CABRERA ET AL.
Serial No. 09/360,542

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 56-95 of the present application are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

In re application of CABRERA ET AL.
Serial No. 09/360,542

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this Amendment, along with transmittal and facsimile cover sheet, are being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. 1.6(d) on the date shown below:

Date: June 24, 2004


Albert S. Michalik

2110 Response to Final